ASCI I/O: MPI-I/O

An API for Parallel I/O

The Scalable Input/Output Project (SI/O) is committed to off-the-shelf solutions and standards where possible. ASCI I/O library writers and application developers are using MPI-IO. The national laboratories are working with vendors to better understand the tradeoffs in designing MPI-I/O libraries.

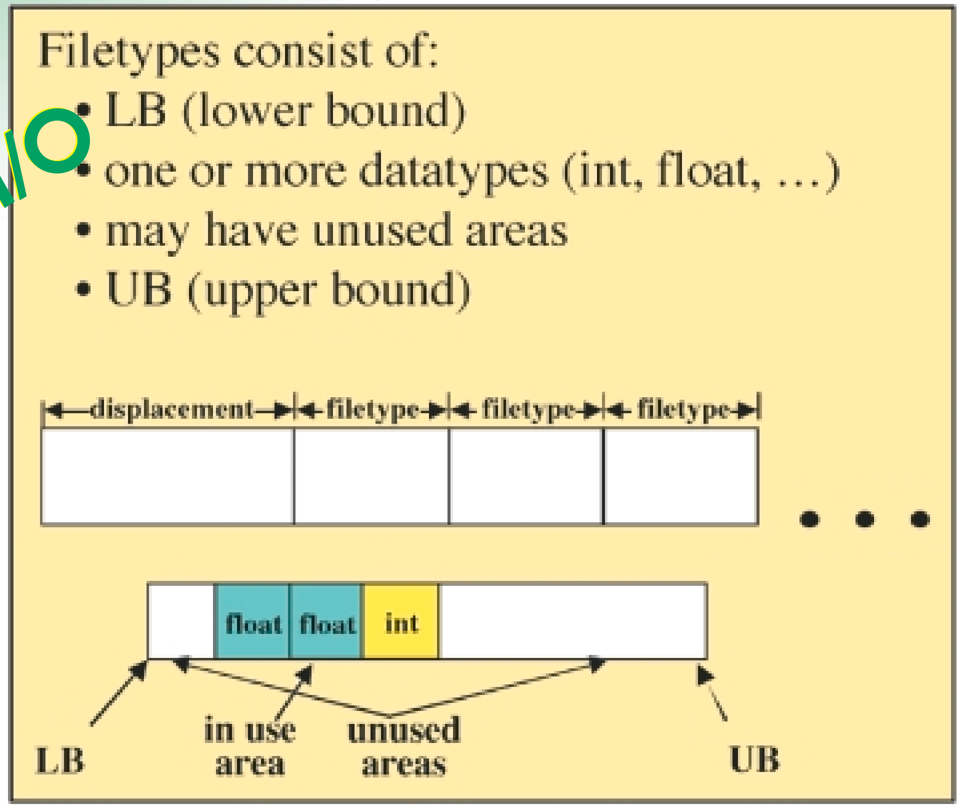

Figure 1: An MPI-I/O file is composed of an initial displacement, followed by a repeated tiling of filetypes.

Motivation For MPI-IO

Parallel applications usually divvy work up among multiple processes or threads. When each process is cooperatively solving a different piece of the same problem, the application designer must decide how to write the results in such a way to preserve data integrity. One way to solve this problem is simply to produce a separate file for each process. Unfortunately, this approach has a number of drawbacks: it requires the user to perform the bookkeeping and management for many related files when one file is preferred; it also may prove inconvenient to visualization tools or other tools which expect the results to be in one file. Since computers are good at bookkeeping and people are not, a better solution is desired.

A second approach is to have the multiple processes write and read from one file. Unfortunately, the most common file system API, POSIX, does not support this activity well. File consistency is maintained on a per-file basis, which means that only one writer can be active at any time to ensure consistency. This solution is also inadequate.

MPI-I/O addresses the need by providing an API that permits applications to express a file in such a way that multiple processes can access a single file in parallel. Further, the file system is given enough information to eliminate overly restrictive consistency constraints.

MPI-I/O Concepts

The basic unit within any MPI file is a filetype. It is simply the template of where data will be placed for each read and write. The filetype may consist of some number of atomic types (for example, floats or integers), or more complicated derived types built up from combinations of derived types (see Figure 1). Each task has its own filetype definition. A typical filetype definition for a four-task application would define 1/4th the filetype as in use, and 3/4ths as void. This way, the filetypes of all four tasks, when merged, completely define all of the bytes within the filetype. The void portion of a filetype is termed a “hole” in MPI-I/O terms.

A view defines the current set of data visible and accessible from an open file. After a file is opened and a view is set, a parallel I/O library can avoid much of the overhead of managing consistency conflicts. The view is able to specify that although multiple nodes are writing to the same file, they never write to the same location (see figure 2).

MPI-I/O includes collective operations. A collective operation is a single operation in which multiple processes take part. For instance, a collective

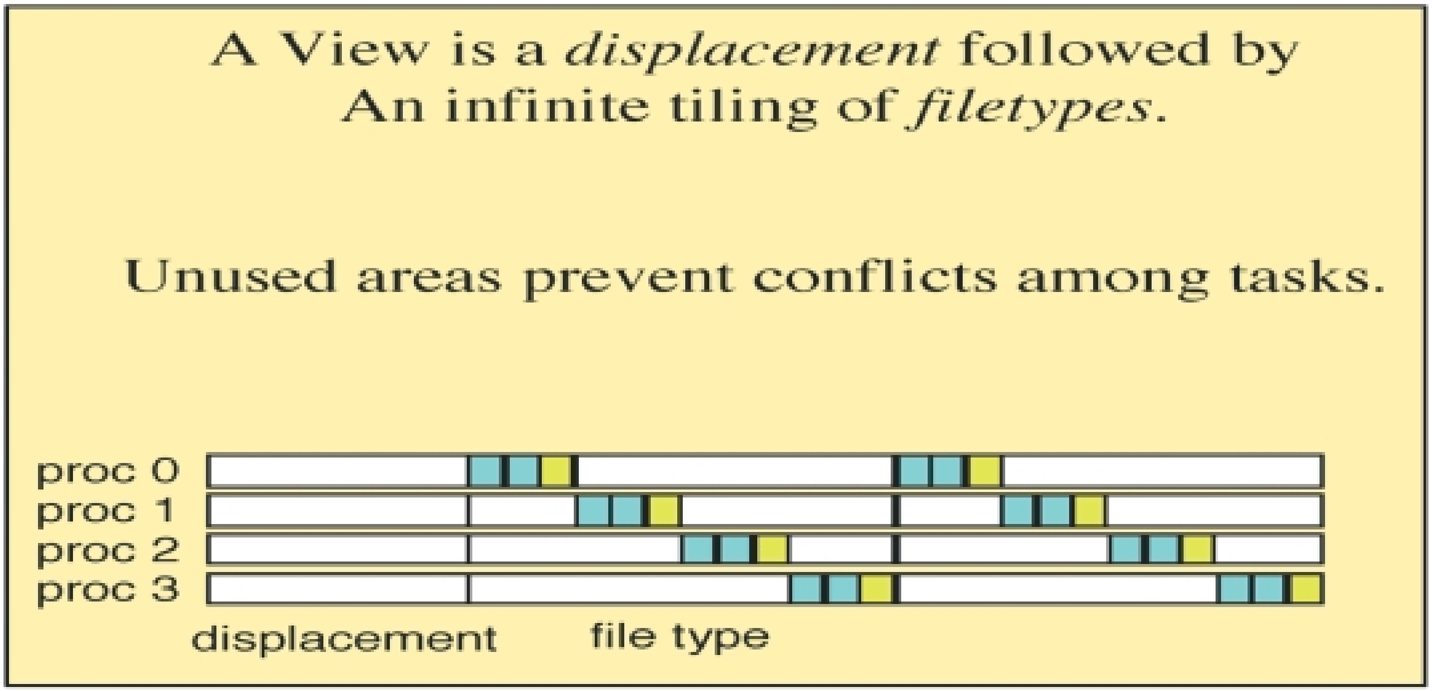

Figure 2: How an entire MPI-I/O view is formed with filetypes .

write occurs when a pre-determined group of processes each participate in a write. Collective operations present the file system with opportunities to improve efficiency.

MPI-I/O supports non-blocking data accesses. A non-blocking access initiates an access, but does not wait for the access to complete. A separate request complete call is used to determine when the transfer has taken place and the application is free to use the buffer.

ASCI Efforts with MPI-I/O

The national laboratories were active in the MPI-Forum which was responsible for the specification of MPI-I/O. Now that MPI-I/O is included with the MPI2 standard specification, efforts are underway to analyze various implementations of MPI-I/O. Further, an effort is underway to develop an ASCI MPI-I/O test suite. This test suite will provide a valuable tool to MPI-I/O implementers which seek to understand how their libraries will be used, and a valuable tool for standardizing performance measurements.

Summary

The ASCI program is developing real applications with MPI-I/O and providing valuable feedback to vendors on their experiences. An ASCI suite of MPI-I/O benchmarks is being developed to help library designers determine which areas present the biggest payoff potential.

For more information on the ASCI Scalable I/O project, please contact:

Judy Sturtevant, (505)845-9448, jesturt@sandia.gov; or

Tyce McLarty, (505) 667-6034, ttm@lanl.gov; or

Kim Yates, (925) 423-5535, rkyates@llnl.gov

Work performed under the auspices of the U.S. DOE by LLNL under contract No. W-7405-ENG-48